



Johnson Space Center
Information Systems Directorate
Information Technology Division

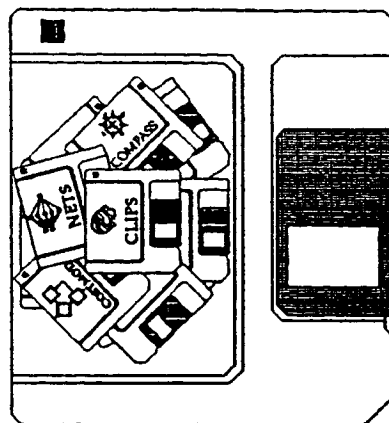
SOFTWARE LIFE CYCLE METHODOLOGIES AND ENVIRONMENTS

Ernest Fridge, Deputy Chief
Software Technology Branch

Mail Code: PT4

Johnson Space Center
Houston, Texas, 77058

August 8, 1991



Software Technology Branch

N92-1735-1

53-61

P-66



SOFTWARE LIFE CYCLE METHODOLOGIES AND ENVIRONMENTS

The software (S/W) development process will be one of the most critical elements of all phases of the Space Station Freedom Program (SSFP), from early design through long-term operations. Improvements in the S/W development process will have significant benefits: reducing both short-term and long-term costs, improving reliability and safety, and improving the functionality and usability of all elements of the Space Station Freedom. This process, though complex, can be improved through the application of a variety of advanced S/W technologies. These new technologies, in the form of methodologies, tool, and environments will benefit both specific Space Station Freedom applications, as well as general Space Station Freedom development practices.

The approach to this activity is to:

- Identify bottlenecks and inefficiencies in existing NASA S/W development practices.
- Evaluate a wide variety of advanced technology approaches for improving the current practices.
- Provide requirements for inserting these new technologies into the SSFP.
- Develop, test, and deploy specific tools, methodologies and environments for use in appropriate places in Space Station Freedom Program.
- Provide assistance in the technology insertion process

The products of this activity will significantly improve the quality and productivity of Space Station software processes by reducing development and maintenance costs, improving software reliability and safety, and broadening the range of problems that can be solved with computational solutions.



Johnson Space Center
Information Systems Directorate
Information Technology Division

OUTLINE/SPEAKERS

Background/Objectives/Benefits - Ernest Fridge

Environments

- CASE - Ernest Fridge
- CLIPS/CLIPS Ada - Gary Riley

Methodologies

- Cooperating Expert Systems - Jorge Rufat-Latre
- Fuzzy Logic - Dr. Robert Lea

Summary - Ernest Fridge



Software Technology Branch



Johnson Space Center
Information Systems Directorate
Information Technology Division

BACKGROUND

The dramatic growth in software in recent years is producing what many writers call the software crisis. More software is required to be produced than the predicted workforce can build. NASA's software projections indicate that they will feel the effect of the crisis unless higher productivity and higher quality can be achieved. NASA's software requirements are increasing drastically with each new program. NASA's systems are extremely large and are both mission and safety critical. In addition, these large systems will last for many years and will be around after the original developers have left the task and after the original technology has become obsolete.



BACKGROUND

- The amount of software to be developed and maintained by NASA is dramatically increasing with each new program
- Very large scale (millions of lines of code) mission critical software systems are to be developed and maintained for many years
- The software development and maintenance process will be one of the most critical elements of all phases of the Space Station Freedom Program (SSFP), from early design through long term operations
- The Shuttle program already faces a staggering software maintenance problem which will be inherited by the SSFP





Johnson Space Center
Information Systems Directorate
Information Technology Division

BACKGROUND (CONT')

The Software Technology Branch (STB) is chartered to develop technologies to combat this crisis. R&D has been underway for several years and techniques to support both conventional and Knowledge Based System (KBS) software have been developed. Coordination with DOD, academia, and commercial tool vendors is being pursued vigorously. The projects discussed in this paper are the ones supported by the SSFP as engineering prototype development, but they leverage the other work done in the STB.



BACKGROUND (CONT')

- The project discussed in this presentation builds upon a base of several years of research into using Knowledge Based Systems approaches to supporting the development and maintenance of both conventional software and knowledge based systems
- The project leverages USAF methodology and environment development research
- The project is part of a larger CASE activity tracking DOD, COTS, NIST standards, and CASE trends plus the development of specific CASE tools





Johnson Space Center
Information Systems Directorate
Information Technology Division

BRANCH SOFTWARE **TECHNOLOGY PROJECTS**

The STB has many advanced software development projects. The projects that are part of the SSFP's engineering prototype development are indicated by an "X". The STB coordinates the effort between these tasks and leverages capabilities as much as possible. This results in individual projects receiving more benefit that would be available from the funding available for each individual task.



Software Technology Branch



Johnson Space Center
Information Systems Directorate
Information Technology Division

BRANCH SOFTWARE TECHNOLOGY PROJECTS

ADVANCED SOFTWARE TOOLS AND ENVIRONMENTS

- Mission and Safety Critical Systems Software Support
- Software Design Recovery and Reuse
- Software Cost Models
- * Engineering Script Languages
- * Advanced Software Development Workstation
- * CASE Tool Consultation/Evaluation/Selection Process
- Knowledge-Based Systems tools
- * Knowledge Acquisition Tools
- * Planning and Scheduling
- * Intelligent Computer-Aided training
- Computational Neural Systems
- Genetic Algorithms
- X-Window tools

("*" - SSFP engineering prototype development)



Software Technology Branch



Johnson Space Center
Information Systems Directorate
Information Technology Division

BRANCH SOFTWARE **TECHNOLOGY PROJECTS**



Software Technology Branch



Johnson Space Center
Information Systems Directorate
Information Technology Division

BRANCH SOFTWARE TECHNOLOGY PROJECTS (CONT')

ADVANCED METHODOLOGIES

- * Fuzzy Logic**
- * Distributed Cooperating Expert Systems**
- Verification and Validation**
- Machine Vision**
- Multi Media**
- Formal Methods**

("*" - SSFP engineering prototype development)



Software Technology Branch



Johnson Space Center
Information Systems Directorate
Information Technology Division

OBJECTIVES

High level objectives are to improve the quality and productivity of the large NASA software development and maintenance projects. Any tool, process, or methodology that improves the software life cycle is a candidate for consideration by the STB. The objectives of this project consider some of the most promising knowledge based systems approaches to support CASE, methodologies, and processes for software reuse, software development, and software maintenance.



OBJECTIVES

- Improve quality and productivity of large software development and maintenance projects through a variety of software technologies such as:
- Computer Aided Software Engineering (CASE)
- Methodologies and Processes for Conventional and Knowledge Based Systems development
- Software Reuse
- Engineering level software application development



BENEFITS

Processes, including software reuse processes, and tools that support and automate those processes seem to be two of the key factors in avoiding unnecessary software costs and in producing higher quality, more reliable, and safer software. The processes, methodologies, and tools being researched and developed within this project effect software across its lifetime. Both conventional and knowledge based systems are addressed. The software lifetime includes the maintenance phase. It also includes the operations phase since the software development process must also include those products such as user guide information required to efficiently operate the software. The size of the NASA systems is also be considered since many processes and tools cannot scale up to support software systems of this large size.





BENEFITS

- Cost avoidance through better processes and the use of better tools
- Improve reliability and safety
- Improve functionality and usability of software elements of SSFP
- Improve software development and maintenance practices
- Provide maintenance and reengineering support to the millions of lines of existing code. Current support is very labor intensive
- Improve the efficiency in operating the complex ground software applications of SSFP





Johnson Space Center
Information Systems Directorate
Information Technology Division

TECHNICAL APPROACH

The STB's charter has goals that include researching and developing methodologies and tools to support improving software engineering of both conventional and knowledge based systems development, operations, and maintenance. The branch's Computer Aided Software Engineering (CASE) outlook insures that the whole software lifecycle gets considered. When the technology is sufficiently mature as shown through proof of concept or other means, it is applied in pilot projects to SSFP elements. The usual activities are the following:

- Identify bottlenecks and inefficiencies in existing NASA software development practices and environments. This requires the STB to keep aware of existing problems and needs within the NASA projects
- Evaluate a wide variety of advanced technology approaches for improving the current practices
- Provide requirements for inserting these new technologies into the SSFP. This is usually done in conjunction with personnel from the application areas
- Develop, test, and deploy specific tools, methodologies and environments for use in appropriate places in SSFP



Software Technology Branch

TECHNICAL APPROACH

Branch charter's goals include researching and developing methodology and tools to support improving software engineering of both conventional and Knowledge Based Systems development, operations, and maintenance.

Technology insertion is provided by applying this technology in SSFP projects

Specific activities include:

- Identify bottlenecks and inefficiencies in existing NASA software development practices and environments
- Evaluate a wide variety of advanced technology approaches for improving the current practices
- Provide requirements for inserting these new technologies into the SSFP
- Develop, test, and deploy specific tools, methodologies and environments for use in appropriate places in SSFP





Johnson Space Center
Information Systems Directorate
Information Technology Division

ENVIRONMENTS



Software Technology Branch



Johnson Space Center
Information Systems Directorate
Information Technology Division

ADVANCED SOFTWARE DEVELOPMENT WORKSTATION (ASDW)

The STB's analysis of CASE product status and direction indicates that a number of good tools are available, but there are some important capabilities and tools that still require further research and development. The Advanced Software Development Workstation (ASDW) project is researching and developing specific types of advanced technology and tools that an advanced workstation for software development should provide. One of these tools is a Parts Composition System (PCS) for developing applications from reusable software parts, using knowledge-based technology. This PCS will have an Engineering Script Language (ESL) as a graphical user interface. This will allow engineers who are not also programmers to generate engineering applications. Another tool being developed is the Intelligent User Interface development Tool (INTUIT). A third tool, a Configurable Control Panel (CCP) for an integrated CASE environment, is being developed under the Framework Programmable Platform (FPP) subtask of the ASDW project. The CCP will be a horizontal tool for managing and enforcing a (locally configurable) model of the software development process.



Software Technology Branch

ADVANCED SOFTWARE DEVELOPMENT WORKSTATION (ASDW)

The ASDW project has been studying ways to apply knowledge based technology to support software engineering. Currently the system has three major components

- The Parts Composition System/Engineering Script Language: Permits an engineer to define an application via a graphical logic diagram. The system contains a library of software parts and the knowledge to support the engineer in populating the graphical diagram with the parts.
- The Framework Programmable Platform: Provides software development process description and work flow control. It follows the Zachman framework concepts and uses the IDEF3 language developed in a joint effort with the USAF.
- The Intelligent User Interface: Provides the support in operating the developed applications. It contains the knowledge of the users guide and advises the user as well as doing constraint testing. Some JSC applications require thousands of inputs and require expert users several weeks to set up and debug.

INTELLIGENT INTEGRATED CASE

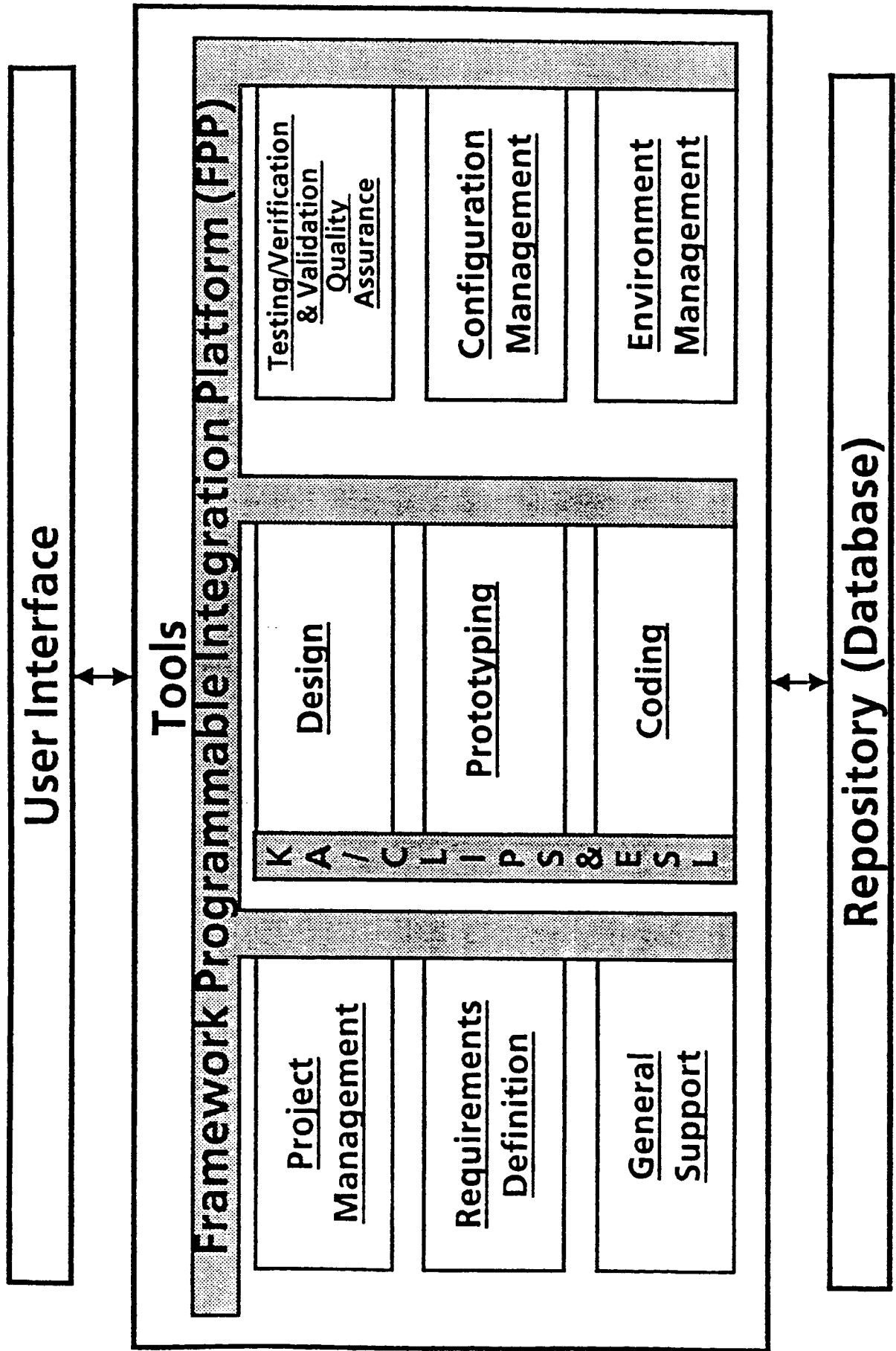
This chart illustrates the role of the software life cycle methodologies and environments task. It illustrates the long term goal of fully integrating all phases of the software lifecycle via the Framework Programmable Platform (FPP) being performed in the Advanced Software Development Workstation subtask. This slide also contains other tools supported by the Software Technology Branch.

The FPP subtask of the ASDW project is researching some of the issues involved in building an integrated CASE environment. The current focus of the FPP is the management and control of the software development and maintenance processes -- crucial factors in the success or failure of any large software system. Specifically, the FPP subtask is developing a horizontal tool, called a Configurable Control Panel (CCP), for specifying, managing, and enforcing a model of the software development process.

The chart also illustrates the near term goal of integrating design, prototyping, and coding via the Intelligent Computer Aided Training (ICAT) work that is merging knowledge acquisition tools with CLIPS to automatically generate expert systems and the use of an Engineering Script Language (ESL) to perform the same function for conventional code.



INTELLIGENT INTEGRATED CASE (CONCEPTUAL OVERVIEW)



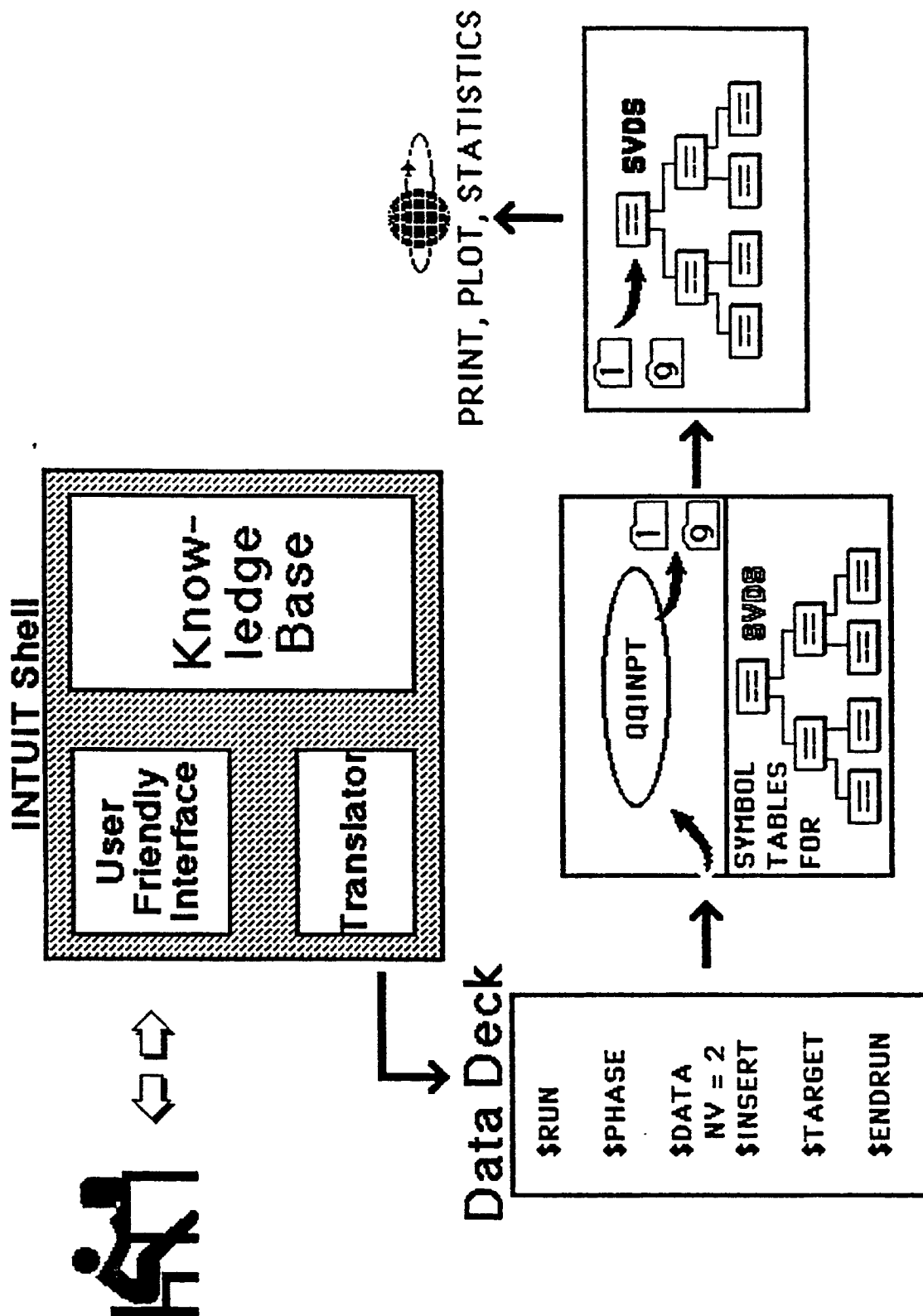
INTELLIGENT USER INTERFACE

A good user interface is critical to the successful use of a complex scientific application such as a space flight simulation, which typically involves very large sets of input data. Even an expert user may expend substantial effort to introduce the right data in the right manner. An Intelligent User Interface (IUI) uses knowledge-based technology to provide the user with the capability to easily prepare the input data without requiring prior extensive knowledge of the underlying software. An IUI is also commonly called a Knowledge-Based Front-End (KBFE). INTUIT (INTElligent User Interface development Tool) is a generic IUI shell that a knowledge engineer configures for a specific application by adding a knowledge base that includes input variable names which are immediately understandable by the users, the range of permissible data values, the structure and format of the data sets, and rules for error and consistency checking. The current knowledge representation scheme used within an INTUIT knowledge base is fully described in. Many of the same subsystems required by a PCS are also required by INTUIT, which may therefore be considered to be a "PCS for input data sets." In fact, INTUIT is a PCS subshell.

The INTUIT shell was used to develop a KBFE for Space Vehicle Dynamics Simulation (SVDS), a computer program currently used at JSC for designing the trajectory and flight plans for Space Shuttle missions. The SVDS application called Ground Simulation (GNDSIM) was selected for KBFE development, and an INTUIT knowledge base was built for it. Flight planners use GNDSIM to verify and refine the sequence of maneuvers required to accomplish a rendezvous. KBFE for GNDSIM can be summarized as follows. All the users who participated in the tests were very satisfied with the KBFE. Building an input data stream with the KBFE proved to require from one-half to one-fifth the time needed using the current interface. As a result of these tests, specific enhancements to INTUIT are planned. The development of KBFEs for other tools used by the flight designers is also being considered.



INTELLIGENT USER INTERFACE FOR SVDS





Johnson Space Center
Information Systems Directorate
Information Technology Division

ASDW POTENTIAL

Most of the activity in the ASDW project is done in a rapid prototyping mode with the future users involved. Field studies with the Shuttle's Flight Analysis and Design System (FADS) in particular has been pursued. The SSFP plans to inherit this software. Personnel from other SSFP ground systems are currently reviewing the three primary ASDW elements.

The ESL/PCS is being field tested to check its applicability to the FADS project on software that could migrate to the SSFP. It has good growth potential since engineering application development from graphical specifications was identified in a JSC survey as a key requirement for future ground and flight applications.

The UI has been evaluated by FADS personnel. The concepts were proved in the FADS project to increase user productivity and were adopted. Growth potential exists since more extensive knowledge based support for constraint testing is being pursued. Even expert users cannot keep in mind the large numbers of constraints that can be violated in runstreams whose inputs number several thousand.

The FPP is still under prototype development but it is getting a lot of attention by possible users. It appears to have good growth potential since NASA is heavily process oriented for producing products of various types. All types of processes conducted by people can be described.



Software Technology Branch

ASDW POTENTIAL

ESL/PCS

- Performing field testing to check its applicability to the FADS project on software that could migrate to the SSFP

GROWTH: Engineering application development from graphical specifications was identified in a JSC survey as a key requirement for future ground and flight applications

IUI

- Concepts were proved in the FADS project to increase user productivity and were adopted.

GROWTH: More extensive knowledge based support for constraint testing is being pursued

FPP

GROWTH: NASA is heavily process oriented for producing products of various types. All types of processes conducted by people can be described.



SLCSE INVESTIGATIVE TASK

The Software Life Cycle Support Environment (SLCSE) was developed for the United States Air Force Rome Laboratory as an advanced development prototype. Its contract was awarded in 1986. It was created to incorporate existing and advanced software engineering technology and was intended to generate documents as a by-product of the software activity. Its framework supports the life cycle phases (requirements, design, etc.), activities (requirements analysis, document production, etc.), roles (project manager, software designer, etc.); and products (software requirements specification, code, etc.)

SLCSE provides an evolutionary foundation for incorporating advances in software engineering technology and tools to support extensibility, tailorability, and scalability. This concept is based upon the unifying life cycle database; formal compilable framework data model that may be tailored; and the CASE tool integration platform to support toolset evolution.

A distinguishing feature of the SLCSE is its underlying entity-attribute/relationship-attribute (EA/RA) model which provides a flexible model capable of supporting a wide range of life cycle phases, activities roles, and products; It facilitates the automated production of documentation allowing capture of necessary information during the course of the project and provides traceability and consistency by storing only one copy of each piece of information and maintaining relationships among entities in order to create life cycle products.





Johnson Space Center
Information Systems Directorate
Information Technology Division

SLCSE INVESTIGATIVE TASK

- Evaluated the framework to see if the concepts could be used to add requirements to current JSC environments
- SLCSE Framework supports:
 - Life cycle phases
 - Activities
 - Roles
 - Products
- Findings and recommendations were provided to the USAF's Rome Laboratory for incorporation into SLCSE enhancements
- Still investigating the information model for its potential on some JSC application developments
- Developed a framework evaluation capability used to evaluate CASE environments. SLCSE contributed significantly to the semantics requirements



Software Technology Branch

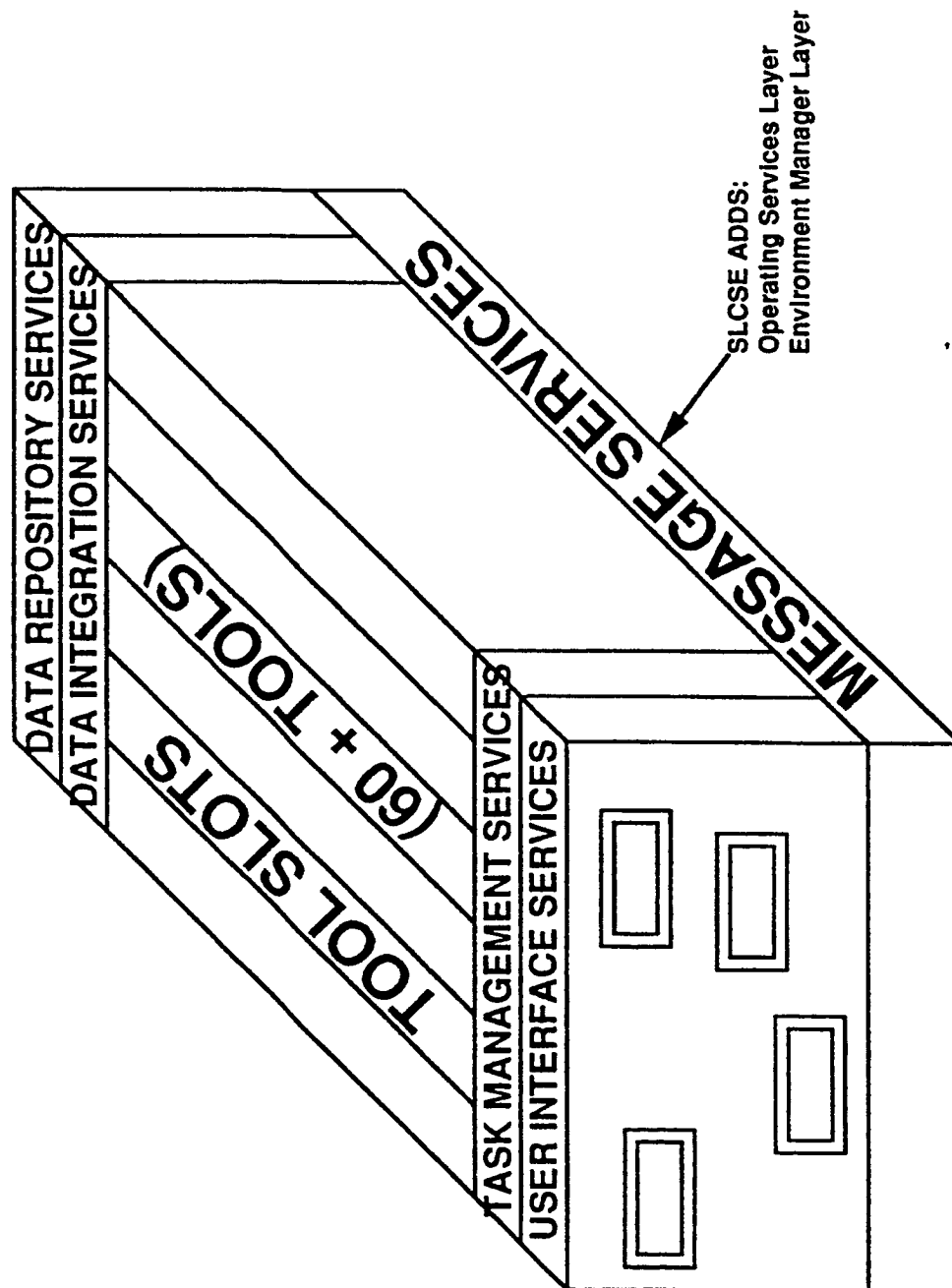
SLCSE/NIST REFERENCE MODEL MAPPING

One of the biggest thrusts in software engineering technology today is the push to develop an open, fully-integrated CASE environment -- i.e., a CASE environment wherein different vendors' hardware and software components can work together effectively. Many of the world's largest computer companies (such as IBM and DEC) and software vendors, the U.S. Department of Defense (DOD), and others are investing great sums in research and development to produce a framework to support an open, fully-integrated CASE environment. A reference model for CASE environment frameworks has been developed by the European Computer Manufacturers' Association (ECMA), and it has been submitted as a proposed standard to the National Institute of Standards and Technology (NIST). This model will be referred to as the NIST Reference Model but it is often informally called the "toaster model" because of its general appearance.

This model is being used by STB so that a consistent metric can be applied to all CASE framework evaluations and recommendations. The SLCSE framework has been mapped to the NIST model and it matches fairly closely. Two additional layers were identified as shown on the chart. The mapping was done by the SLCSE developers.



SLCSE/NIST RM MAPPING





Johnson Space Center
Information Systems Directorate
Information Technology Division

SLCSE POTENTIAL

The current version of SLCSE is a prototype. The concepts are excellent and can be used to evaluate framework products and CASE environment. It maps very well with the NIST reference model for CASE environments. The information model is useful as it stands. STB plans to investigate the use of the current prototype more and to follow the development of the commercial product.



Software Technology Branch



Johnson Space Center
Information Systems Directorate
Information Technology Division

SLCSE POTENTIAL

- The information model of the DOD's 2167 software development process is perhaps the best available
- SLCSE is consistent with the NIST reference model for CASE framework
- The current planned enhancements are intended to turn the current prototype model into a robust framework for supporting mission critical software development



Software Technology Branch



Johnson Space Center
Information Systems Directorate
Information Technology Division

CASE TECHNOLOGY SELECTION AND INSERTION PROCESS

Technology transfer of emerging CASE technology to NASA is the primary objective of all of the STB's CASE projects. Technology transfer can occur in many different ways, including the direct use of tools developed by the STB, the adoption of new methods and technology identified by the STB (such as repository technology when it matures), and the purchase of appropriate commercial off-the-shelf (COTS) tools.

The STB set a goal of collecting information about existing CASE products and characterizing those products in order to provide a CASE tool consulting service for the JSC community. The motivation for establishing such a service was the complex nature of CASE and the confusing status of the CASE tool market. The sheer number of available CASE tools and the rapid rate of change of the CASE market, coupled with unrealistic consumer expectations of what CASE tools can do, have led to some exaggerated claims about CASE and, consequently, to some disappointed consumers.

Rather quickly, the scope of this CASE consulting service was expanded to a software engineering consulting service. CASE tools can only assist in the software engineering process. A software development organization must have a well-managed, repeatable, and explicit software development process. If a disciplined software engineering process does not exist within an organization, then that organization must adopt one, and this will likely imply a change in its way of doing business.

Technology insertion, in particular, seems to be successful only when key people within the organization (called change agents) actively participate. Change agents tend to be knowledgeable persons in middle management, recognized for their abilities and credibility by both upper and lower management.



Software Technology Branch

CASE TECHNOLOGY SELECTION

AND INSERTION PROCESS

- CASE environments are complex in nature
 - No single vendor will assume total investment risk
- Hundreds of CASE vendors are marketing tools and the tool market is rapidly changing
- Tools and environments will change the development and maintenance culture and drastic changes will result in the tools not being used
- A process is required before tools are selected
- The tool environment should support the culture and the process
- A CASE technology insertion process has been developed to help organizations utilize CASE





Johnson Space Center
Information Systems Directorate
Information Technology Division

CASE TECHNOLOGY SELECTION AND INSERTION PROCESS

In order to make sound CASE recommendations and to improve the chances of achieving CASE technology insertion, there are five basic activities that occur during the process: characterize the organization's culture; characterize the software systems produced; identify improvements to the organization's software engineering process; identify candidate tools and environments; and develop a technology insertion plan. The process is somewhat iterative as most software processes are. Some activities can proceed in parallel as shown on the chart.

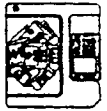
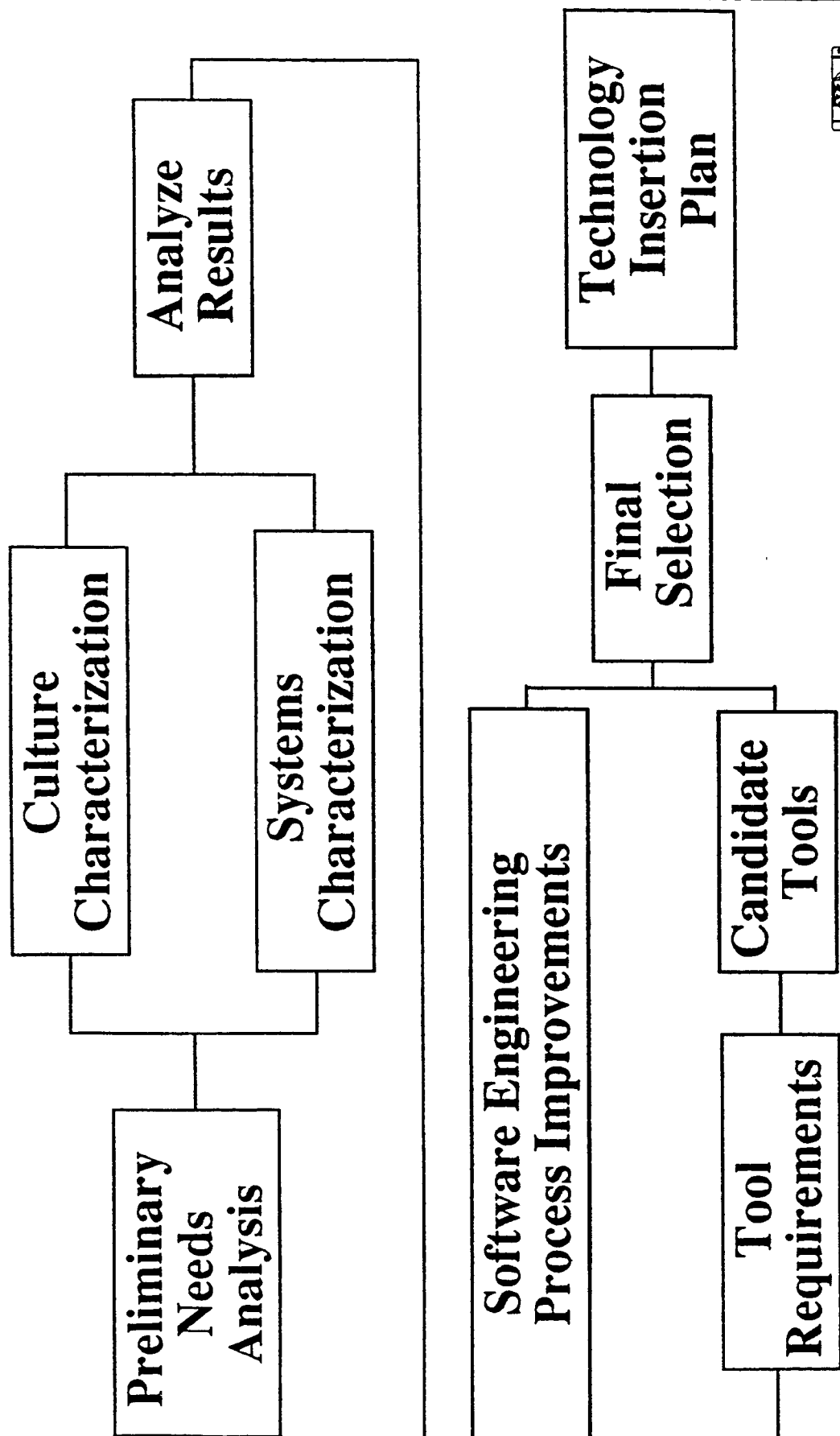


Software Technology Branch



Johnson Space Center
Information Systems Directorate
Information Technology Division

Technology Selection and Insertion Process



Software Technology Branch



Johnson Space Center
Information Systems Directorate
Information Technology Division

CASE PROCESS POTENTIAL

This task has produced considerable interest at JSC. It is being used for the first time to help the Information Systems Directorate select CASE tools and environments for JSC's institutional information systems. This work will certify the process. Another task has just been initiated to help the Mission Operations Directorate select CASE tools for both the SSFP ground support systems and the Shuttle maintenance system. The system will also be used with the Strategic Avionics Technology Working Group (SATWG)'s effort to determine the CASE requirements to support future generic avionics software development.



Software Technology Branch



Johnson Space Center
Information Systems Directorate
Information Technology Division

CASE PROCESS POTENTIAL

CURRENT

- Process is being certified by applying it to JSC institutional information systems
- Process is being initiated to apply it to the ground operations and flight planning systems for SSFP as a pilot project

GROWTH

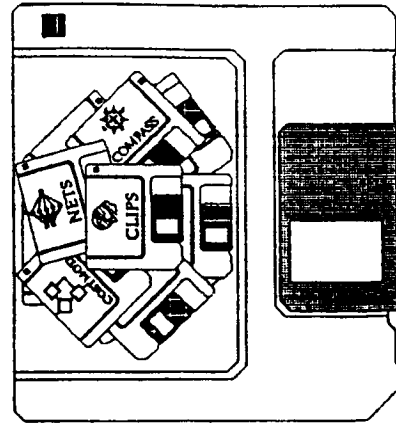
- The process will be used by the SATWG software subcommittee to determine the CASE requirements for supporting future SSFP maintenance and the exploration program



CLIPS and CLIPS/Ada

Task Managers: Chris Culbert and Gary Riley
NASA/Johnson Space Center
Software Technology Branch

Presented by: Gary Riley





Johnson Space Center
Information Systems Directorate

CLIPS and CLIPS/Ada

Expert systems are computer programs which emulate human expertise in well defined problem domains. The potential payoff from expert systems is high: valuable expertise can be captured and preserved, repetitive and/or mundane tasks requiring human expertise can be automated, and uniformity can be applied in decision making processes.

The C Language Integrated Production System (CLIPS) is an expert system building tool, developed by the Software Technology Branch at the Johnson Space Center, which provides a complete environment for the development and delivery of expert systems. CLIPS was specifically designed to provide a low cost option for developing and deploying expert system applications across a wide range of hardware platforms. The use of CLIPS has many benefits: CLIPS runs on conventional hardware systems and is completely portable to a wide range of computers; CLIPS can be integrated with and embedded within conventional software systems; CLIPS source code is free to all government agencies.; CLIPS can be easily extended and modified; CLIPS can be used with environment specific interfaces for PC compatible, Macintosh, and X Window environments; CLIPS comes with extensive documentation; and CLIPS users can receive support from either a help desk or an electronic bulletin board.

An version of CLIPS developed entirely in Ada, CLIPS/Ada, is also available for use.



Software Technology Branch

CLIPS and CLIPS/Ada

- Expert systems are computer programs which emulate human expertise in well defined problem areas.
- The C Language Integrated Production System (CLIPS) is a programming language environment used for the creation of expert system applications.
- CLIPS was specifically designed to provide a low cost option for developing and delivering expert system applications across a wide range of hardware platforms.
- CLIPS/Ada is a version of CLIPS developed entirely in Ada.



Approach

The current release of CLIPS, version 5.0, provides support for rule-based, object-oriented, and procedural programming. Rule-based programming allows knowledge to be represented as heuristics which specify a set of actions to be performed for a given situation. Object-oriented programming allows complex systems to be modeled as modular components (which can be easily reused to model other systems or to create new components). Procedural programming allows a set of instructions to be grouped together in a procedure which examines or manipulates data.

The current release of CLIPS/Ada, version 4.3, provides support for rule-based programming and is fully syntax compatible with version 4.3 of CLIPS. The next planned release of CLIPS/Ada, version 4.4, is scheduled for September 1991 and will support rule-based and procedural programming.



Approach

- The current release of CLIPS, version 5.0, provides support for rule-based, object-oriented, and procedural programming.
- The current release of CLIPS/Ada, version 4.3, provides support for rule-based programming and is fully syntax compatible with version 4.3 of CLIPS.
- The next planned release of CLIPS/Ada, version 4.4, is scheduled for September 1991 and will support rule-based and procedural programming.



SSFP Integration

Baseline Program

Space Station Freedom applications will require deep integration of expert system technology with applications developed in conventional languages, specifically Ada. Since SSF has a requirement that all SSF software be developed in Ada, Ada based expert system tools may be required to allow the use of expert systems in SSF applications. The ability to apply automation to SSF functions could be greatly enhanced by widespread availability of state-of-the-art expert system tools based on Ada. At a minimum, Ada based tools will ease integration issues for expert systems used in SSF applications. Although there have been some efforts to examine the use of Ada for expert system applications, there are few existing products which provide state-of-the-art capabilities in an Ada tool. The development of CLIPS/Ada version 4.3 has been completed and this version is ready to meet SSFP Ada requirements. Both CLIPS and CLIPS/Ada are being distributed by the SSE.

Growth & Evolution

The continued growth and evolution of the Space Station will require extensive automation to reduce operational costs and manpower requirements as well as enhance safety and reliability. The use of expert systems will provide an effective means for developing the high levels of automation required. Several SSFP contractors are already using CLIPS to evaluate advanced automation concepts for SSFP evolution.



SSFP Integration

Baseline Program

- CLIPS/Ada is developed and ready to meet SSFP Ada requirements.
- CLIPS and CLIPS/Ada are being distributed by the SSE.

Growth & Evolution

- The continued growth and evolution of the Space Station will require extensive automation to reduce operational costs and manpower requirements as well as enhance safety and reliability.
- Several SSFP contractors are already using CLIPS to evaluate advanced automation concepts for SSFP evolution.





Johnson Space Center
Information Systems Directorate
Information Technology Division

METHODOLOGY



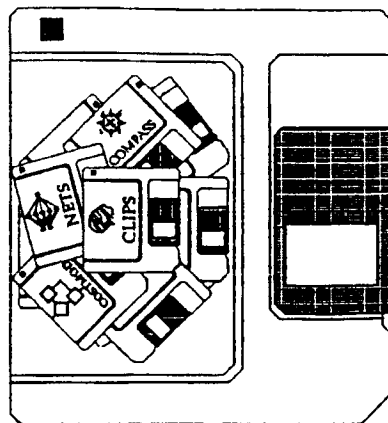
Software Technology Branch

Distributed Intelligent Systems

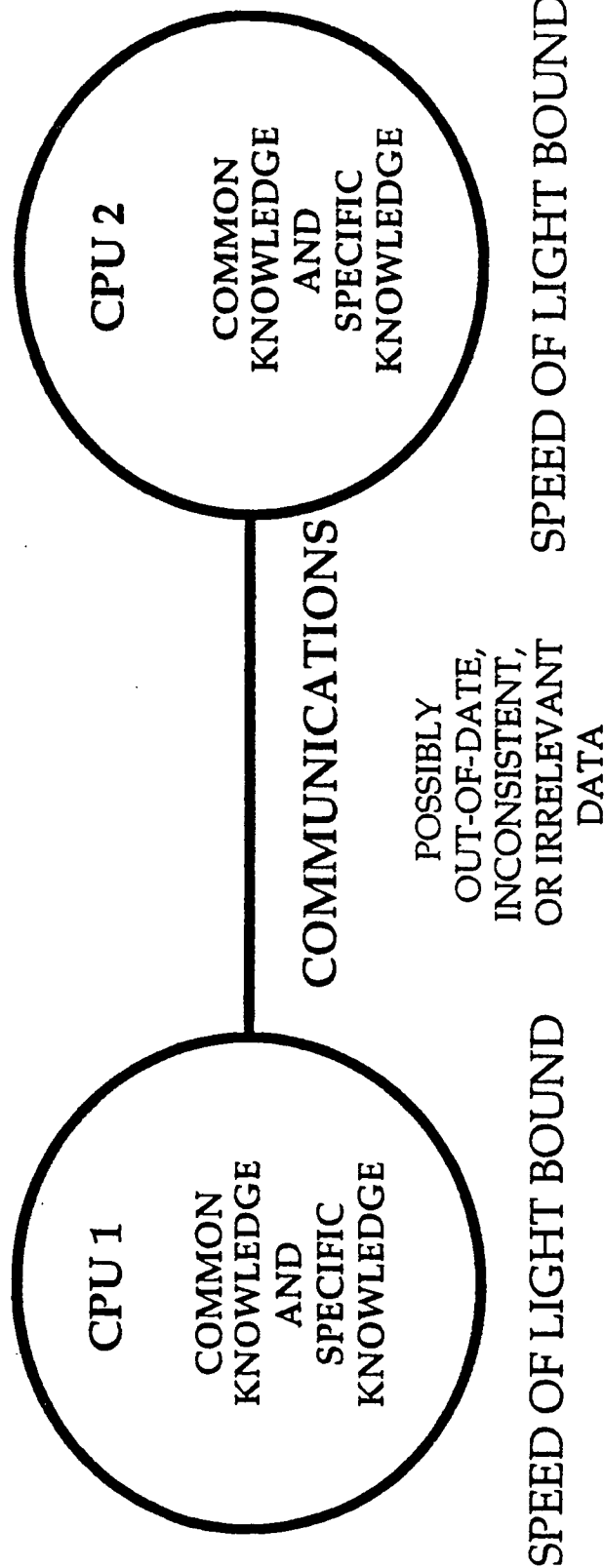
Task Manager: Chris Culbert
NASA/Johnson Space Center
Software Technology Branch

Task Support: Jorge Rufat-Latre (MDSSC)
Grayum Watts (MDSSC)
Paul Vu (MDSSC)
Brooks Slaughter (MDSSC)

Presented by: Jorge Rufat-Latre (task lead)



The need for Cooperation



COOPERATION BETWEEN LOOSELY COUPLED AGENTS
IS AN EFFECTIVE MECHANISM TO OVERCOME
PHYSICAL BOUNDS ON COMPUTING POWER



Distributed, Intelligent Systems Objectives

Requirement

- As computer applications expand in complexity and scope, it becomes increasingly important to take full advantage of all available processing resources. Improvements in processing speed, drastic reductions in workstation prices, and the spread of local area networks have allowed mission control center applications to move from mainframes environments to widely distributed workstation environments. Similar improvements in space qualified hardware systems will eventually lead to distributed computer systems in spacecraft also. However, to take full advantage of these environments requires the development of mechanisms for sharing and coordinating distributed information.

Solution

- Intelligent data distribution and distributed scheduling are two examples of new capabilities achieved by applying these state-of-the-art technologies. Peer-to-peer intelligent agent cooperation and negotiation schemes coupled with rigorous performance studies open new areas of opportunity. Properly designed and implemented systems which can work cooperatively in distributed processing environments can significantly improve system reliability by improving fault tolerance. They can also provide significant performance enhancements and improve design flexibility. Finally, they provide an effective environment for solving more complex problems than those which can be addressed by non-distributed systems.



Distributed, Intelligent Systems Approach

The basic approach for this task is to:

- Develop techniques for using distributed, cooperative, intelligent systems to support management of ground or on-board systems. Demonstrate these capabilities in appropriate applications.
- Integrate multiple intelligent systems in a distributed processing testbed and demonstrate different approaches to distribution and cooperation.
- Analyze the various trade-offs and design decisions associated with varying levels of cooperation and intelligence in on-board or ground systems.



Distributed, Intelligent Systems

SSFP Integration

Baseline Integration

The Real Time Data System (RTDS) is currently used in the Mission Control Center to support Shuttle missions. This system has proven invaluable for allowing the development and use of advanced systems to support ongoing Shuttle operations. One of the objectives of this task, to be performed under the direction of Troy Heindel, is to provide significant improvements in RTDS functionality by making RTDS data available to numerous application workstations across a LAN. Another goal is to improve the performance and fault tolerance of the RTDS by appropriately designing and developing distributed communications capabilities. Short term plans will include the development of appropriate communications protocols and communications managers for distributing RTDS data to multiple workstations. Long term plans will evaluate approaches to long-term communications and cooperation needs, including looking at multicast or broadcast options, negotiation based managers, and other approaches to providing large amounts of RTDS data to multiple applications in real-time. These will be required to allow use of RTDS in Space Station Freedom applications.

Growth and Evolution

As the Space Station Freedom matures, it will become increasingly important to share and manipulate information residing in multiple locations. Ongoing work at NASA/LeRC has developed approaches to this problem based on price/cost models for scheduling of activities. This approach is similar to the negotiation approaches to cooperation. Their current approach is hierarchically structured and not very distributed. During FY92, we will work with NASA/Lewis to define peer-to-peer approaches to negotiation based cooperation (a more robust, fault tolerant technique than pure hierarchies) as well as developing better ways of using these techniques in a distributed environment. Finally, we will try to apply these techniques to distributed versions of COMPASS (a NASA scheduling tool), and possibly to future (long term) versions of RTDS.



Distributed, Intelligent Systems SSFP Integration

Baseline Program

- Support the development of distributed processing tools for the current RTDS in Mission Control Center. Evaluating advanced approaches to distributed, cooperative systems for use in evolution of RTDS for Space Station support. These extensions will improve the efficiency and fault tolerance of RTDS when used in the high data volume environments typical for Space Station Freedom.

Growth & Evolution

- In conjunction with ongoing work at NASA/Lewis, develop approaches to distributed processing which support advanced applications such as distributed scheduling. Such systems will become increasingly important as SSF evolves and require use of advanced algorithms for effective use of SSF computing resources.





Johnson Space Center
Information Systems Directorate

Fuzzy Control Applications for Space Station

Task Manager:

Robert N. Lea
NASA / Johnson Space Center
Software Technology Branch

Task Support:

Jack Aldridge (MDSSC)
Jonathan Weiss (MDSSC)
Yashvanti Jani (LinCom)
James Villarreal (NASA - PT4)

Presented by:

Robert N. Lea



Fuzzy Logic

Fuzzy logic is a logic based on Lotfi Zadeh's theory of fuzzy sets. Fuzzy sets can be used to represent ambiguous, vague, or imprecise conditions. They are defined by continuous functions from the universal set to the unit interval thus generalizing the notion of a characteristic function of a set. This concept permits evaluation of the degree to which a statement or condition is satisfied. Thus Zadeh's logic is concerned with the formal principals of approximate reasoning, the logic that allows one to infer approximate answers to questions whose premises and conclusions are conditions that may change continuously from true to false, rather than abruptly, and therefore may sometimes be only partially true.

The type of uncertainty that fuzzy logic deals with is different from the uncertainty dealt with by probabilistic based methods. The probability that a random sample will belong to a set is different from the degree to which the sample belongs to the set. For example, if a person walks into a room an evaluation can be made as to how well the person fits the condition of being tall. This is fuzzy uncertainty since the set of tall people is not Boolean. On the other hand if people are labeled as tall or not tall this is not a fuzzy or probability problem since the person is there and the conclusion is known. The underlying premise is that fuzzy logic is not a competing method with probabilistic methods for handling uncertainty, but a method of handling uncertainty of a different type. Specifically, fuzzy logic deals with evaluations of degrees to which certain conditions have occurred as opposed to predictions as to whether or not they will occur.

Fuzzy logic is therefore a natural concept to apply to expert systems development for modelling the rules of the expert that invariably are stated in natural language which is inherently full of fuzzy terms.



Fuzzy Logic

FUZZY LOGIC IS A LOGIC OF COMMON SENSE REASONING THAT

- ALLOWS FOR PARTIAL TRUTHS OF CONDITIONS, e.g. *LOW* TEMPERATURES, *SMALL* RESIDUALS
- PROVIDES A METHOD FOR SOLVING ILL-DEFINED AND COMPLEX NON-LINEAR CONTROL PROBLEMS BASED ON DECISION MAKING AND SENSOR EVALUATION LOGIC LEARNED FROM HUMANS
- INCREASES THE EFFECTIVENESS OF RULE BASED EXPERT SYSTEMS IN MODELING NATURAL LANGUAGE STATEMENTS OF RULES, e.g., IF TEMP IS *HIGH* AND RPM IS *FAST* THEN PRESSURE SHOULD BE *REDUCED*.
- PROVIDES A METHODOLOGY FOR EVALUATING THE UNCERTAINTY OF VAGUENESS OF NATURAL LANGUAGE EXPRESSION AS OPPOSED TO PREDICTING THE PROBABILITY OF CORRECTNESS



MOTIVATION FOR SPACE STATION APPLICATIONS

Our decision to pursue fuzzy logic for space station applications has been motivated to a large extent by early successes at the JSC in the development of fuzzy logic methodologies for vehicle and process control and decision making. However, one of the strong driving forces has been the outstanding successes of the Japanese in the commercial development of fuzzy logic applications over the last three or four years. These include applications in high technology areas such as train control systems, camera stabilization and autofocusing systems, automotive applications to automatic transmission and braking systems, air conditioning control systems, television auto contrast and brightness control, and control of nuclear reactors as well as commercially successful applications to household products such as washing machines and vacuum cleaners.



MOTIVATION FOR SPACE STATION APPLICATIONS

DECISION TO PURSUE FUZZY LOGIC APPLICATIONS FOR SPACE STATION APPLICATIONS HAS BEEN BASED ON

- MANY SUCCESSES OF FUZZY LOGIC APPLICATIONS IN JAPAN
 - SENDAI SUBWAY SYSTEM
 - CAMERA STABILIZATION AND AUTOFOCUSING
 - AIR CONDITIONING CONTROL SYSTEMS
 - AUTO TRANSMISSION AND BRAKING CONTROL
 - TELEVISION AUTO CONTRAST AND BRIGHTNESS CONTROL
- JSC SUCCESSES IN SIMULATED SPACE VEHICLE CONTROL, PROCESS CONTROL, TETHER CONTROL, AND OTHER APPLICATIONS



Selection of Applications

NASA HQ Level I co-sponsored (with McDonnell Douglas Space Systems) a Workshop in Fuzzy Logic Control for Space Station Applications. The purpose of the conference was to identify potential applications of Fuzzy Logic and to determine the validity of those applications. Attendees included internationally recognized experts in Fuzzy Logic (Kosko, Yen, Togai, Xu, Sugeno, Lea, Ma, Berenji, Jani), NASA managers (Gersh, Fernquist, Lea, Lawler), and MDSSC managers and engineers with knowledge of the potential Space Station applications. While all problems were judged to be appropriate applications of Fuzzy Logic, some were selected for immediate efforts based on considerations of timing, NASA priorities, and potential spin-off opportunities.



Selection of Applications

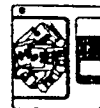
- **11 Candidate applications identified at November 1990 Huntington Beach Workshop**
 - **problems described by engineers**
 - **initial responses from noted fuzzy experts**
 - **initial screening based on technical merit, NASA need, and timing factors**
- **Final candidates selected based on**
 - **probability of baseline application**
 - **immediate NASA benefit**
 - **committed ownership by user**
 - **potential for multiple applications**
 - **low risk and cost**



Applications - Blowdown Thruster

Space Station reboost will be accomplished using blowdown thrusters (engines whose thrust comes from gas pressure which decreases with usage). Guidance, Navigation, and Control needs to predict accurately the level of thrust available at different points throughout the maneuver, but this is complicated because the same engines provide thrust for attitude control, resulting in rapid fluctuations in the actual usage curves.

Fuzzy Logic can be used to develop a rule-based time series model that fits blowdown performance better than a polynomial approximation. Even more value may be achieved if a fuzzy control model can be developed to measure and improve the efficiency of attitude controls during the maneuver (e.g. by timing x-direction thrusts to minimize the need for y- and z-corrections).



Applications

- Blowdown Thruster - difficult to predict thrust levels as engine blows down due to unknown rate of attitude control thrusting
- Current status - work order started 7/22/91; have identified two promising fuzzy approaches and met with GN&C experts to validate approach





Applications - RFMD Pump

The Rotary Fluid Management Device Pump may flood due to heat load changes on the evaporators in the Thermal System. When this happens, current controls react with large transient power corrections. These in turn place unnecessarily high loads on electrical components and may use power inefficiently.

By using Fuzzy Logic Control instead of proportional control, we may be able to spread a more gradual response over a longer period of time, thus reducing power transients and potential overshoots. A fuzzy rule-based control system incorporate timing and safety constraints while producing this smoother response and utilize power more efficiently.



Applications

- RFMD Pump - pump may flood due to heat load changes on evaporators; constant frequency control leads to excessive power excursions; by matching pump RPM to thermal load, may be able to minimize power usage
- Current status - work order started 7/22/91; initial approach to flooding problem developed





Johnson Space Center
Information Systems Directorate
Information Technology Division

SUMMARY

The software methods and environments being researched and developed within the STB will make CASE technology available for both conventional and knowledge based systems. The support extends across the development, operational, and maintenance phases of the life cycle. This will significantly improve both the quality and productivity of SSFP software and will avoid cost resulting from poor quality software and poor processes. The reliability and safety of the software will be improved. Tasks are directed towards the environment and tools and towards methodologies for providing advanced software solutions.



Software Technology Branch

SUMMARY

- Products of this project will significantly improve the quality and productivity of SSFP software processes by:
 - Improving software reliability and safety
 - Broadening the range of problems that can be solved with computational solutions
- Project brings in CASE technology for:
 - Environments
 - ESL/PCS application generator
 - Intelligent User Interface for cost avoidance in setting up operational computer runs
 - Framework programmable platform for defining process and software development work flow control
 - Process for bringing CASE technology into an organization's culture
 - CLIPS/CLIPS Ada language for developing expert systems





Johnson Space Center
Information Systems Directorate
Information Technology Division

SUMMARY (CONT')

- Methodologies
- Method for developing fault tolerant, distributed systems
- Method for developing systems for common sense reasoning and for solving expert systems problems when only approximate truths are known

